Project Report

Department of Electrical engineering
Sharif University of Technology

Name:	Mohammad Mahdi Sharifian
Student Number:	403203462
Research Centre:	Sharif University of Technology
Research Project Title:	PPM-4 Error Probability: classical photon counting Vs quantum detection
Primary Supervisor:	Dr Javad Salehi

Project Summary

This project benchmarks the performance of 4-slot pulse-position modulation (4-PPM) on a noisy optical channel using two detection paradigms. The **classical arm** models an ideal photon-counter receiver: counts in the pulse slot follow a Poisson law with mean $N_s + \mathcal{N}$ while the three empty slots carry background noise $\mathcal{N}$; an exact closed-form series yields the symbol-error probability. The **quantum arm** implements the square-root measurement (SRM) prescribed by quantum-detection theory: displaced-thermal single-slot states are truncated to a 30-dimensional Fock basis, combined into 6-rank square-root factors, and processed through the Gram-family formalism to obtain the SRM success probability. Sweeping $N_s \in [0, 8]$ photons at four noise levels $\mathcal{N} = \{0, 0.05, 0.10, 0.20\}$, the code reproduces Cariolaro's Fig. 8-23: four green curves for the classical receiver and four black/grey curves for the quantum SRM, confirming a 2–3-order-of-magnitude error-rate advantage of quantum detection across the entire operating range.

1 Classical PPM4 Detection

Introduction

Pulse–position modulation (PPM) encodes information in the temporal location of a short optical pulse within a fixed symbol frame. In the four–slot case (4-PPM) each symbol carries two bits, and the receiver must decide which of the four slots contains the pulse. In the *classical-detection* paradigm studied in this section the front-end is an *ideal photon counter*; for every symbol it yields four non-negative integers $\{X_0, X_1, X_2, X_3\}$ representing the photon counts registered in the pulse slot and in the three "empty" slots. Owing to shot noise and background radiation these counts are modelled as independent Poisson random variables: the pulse slot follows $\text{Pois}(N_s+\mathcal{N})$ while each empty slot follows $\text{Pois}(\mathcal{N})$, where N_s is the mean signal energy and $\mathcal{N}$ the mean background energy per slot. A symbol is decoded *correctly* only if the pulse slot count is *strictly larger* than every empty-slot count; ties are conventionally treated as errors. The resulting symbol-error probability P_e—a function of $(N_s, \mathcal{N})$—serves as the classical reference against which the quantum square-root measurement (SRM) will be benchmarked in the next section. Here we derive an exact closed-form series for P_e, present a numerically robust evaluation algorithm, and visualise how classical 4-PPM performance degrades as the background level increases.

Mathematics

Poisson statistics. Let X denote the photon count in the pulse slot and Y_1, Y_2, Y_3 the counts in the three empty slots. Under the classical-photon–counter model these random variables are independent and distributed as

$$X \sim \text{Pois}(N_s + \mathcal{N}), \qquad Y_i \sim \text{Pois}(\mathcal{N}),\ i = 1, 2, 3.$$

For a Poisson mean μ we denote the probability–mass function and lower cumulative distribution function by

$$p_\mu(n) = \frac{\mu^n e^{-\mu}}{n!}, \qquad F_\mu(n) = \Pr\{Y \le n\} = e^{-\mu} \sum_{k=0}^{n} \frac{\mu^k}{k!}.$$

Symbol correctness. A symbol is decoded correctly iff the pulse slot registers a strictly larger count than *every* empty slot, i.e. $X > Y_1,\ X > Y_2,\ X > Y_3$. Conditioned on the event $\{X = n\}$, the probability that all empty slots are $< n$ is $[F_{\mathcal{N}}(n-1)]^3$. Averaging over X gives the exact success probability

$$P_c^{class}(N_s, \mathcal{N}) = \sum_{n=0}^{\infty} p_{N_s+\mathcal{N}}(n)\ \big[F_{\mathcal{N}}(n-1)\big]^3, \tag{1}$$

with $F_{\mathcal{N}}(-1) := 0$ to cover the $n = 0$ term. The symbol–error probability is then

$$P_e^{class}(N_s, \mathcal{N}) = 1 - P_c^{class}.$$

Tail truncation. Numerical evaluation of eq:Pc$_c lassicrequiresafinitecut - off$n$_{\max}$. For a chosen tail mass $\varepsilon = 10^{-18}$ we pick

$$n_{\max} \;=\; \max\big\{n : \Pr\{X > n\} \le \varepsilon,\ \Pr\{Y > n\} \le \varepsilon\big\} \;+\; 2,$$

where the inequalities are inverted with the Poisson *inverse-survival function*[1]. If the library routine overflows we fall back on the analytic bound $n_{\max} \approx \mu + 12\sqrt{\mu} + 10$. With this truncation the neglected probability mass is below ε and the resulting P_e is accurate to at least 10^{-18}.

Complexity. Equation eq:Pc$_c lassicbecomesadotproductoftwo(n_{\max}+1)$-vectors and is therefore $\mathcal{O}(n_{\max})$; for the worst case ($N_s = 8,\ \mathcal{N} = 0.2$) we have $n_{\max} \approx 60$, so the entire N_s–$\mathcal{N}$ grid is evaluated in a few milliseconds on a laptop.

Code Structure

The Python implementation[2] is organised into four logical blocks, each mapped to a sub-routine that mirrors the mathematical flow of Sec. **??**.

1. **Robust Poisson–tail truncation**

   `_poisson_trunc(mu, tail)` Inverts the Poisson survival function to find the smallest integer $n_{\max}$ such that $\Pr\{N > n_{\max}\} \le$ `tail`. A safety expression $n_{\max} \approx \mu + 12\sqrt{\mu} + 10$ is used if SciPy overflows or returns ∞; this keeps the routine numerically robust for all $(N_s, \mathcal{N})$ pairs swept in the study.

2. **Exact error-probability kernel**

   `classical_pe(Ns, Nbkg)` Implements Eq. eq:Pc$_c lassicby$

   building a vector $\big\{p_{N_s+\mathcal{N}}(n)\big\}_0^{n_{\max}}$,

   building $\big\{[F_{\mathcal{N}}(n-1)]^3\big\}_0^{n_{\max}}$,

   returning $P_e = 1 - \langle \mathbf{p}, \mathbf{F} \rangle$.

   Both vectors are NumPy arrays; the inner product is a single call to `np.dot`, giving each evaluation $O(n_{\max})$ complexity.

Parameter sweep

`sweep_pe(Ns_grid, N_list)` Loops over the user-defined photon grid $\{N_s\}$ and noise list $\{\mathcal{N}\}$, storing results in a dictionary `{Nbkg : [Pe(Ns)]}` that is directly plottable. Typical grid (17 points × 4 noise levels) completes in $\mathcal{O}(10^{-3})$ seconds.

Plotting routine

Main block Converts the dictionary to four green solid curves, darkest for $\mathcal{N} = 0$ and progressively lighter as noise increases. The `matplotlib` call stack is kept identical to the quantum script so the two plots can later be merged seamlessly.

This modular design lets the numeric kernel be unit-tested in isolation and reused inside Monte-Carlo simulations or optimisation loops, while the plotting layer remains a thin, easily customisable wrapper.

[1] 'scipy.stats.poisson.isf' in the code.

[2] `ppm_classical_plot.py` in the project repository.

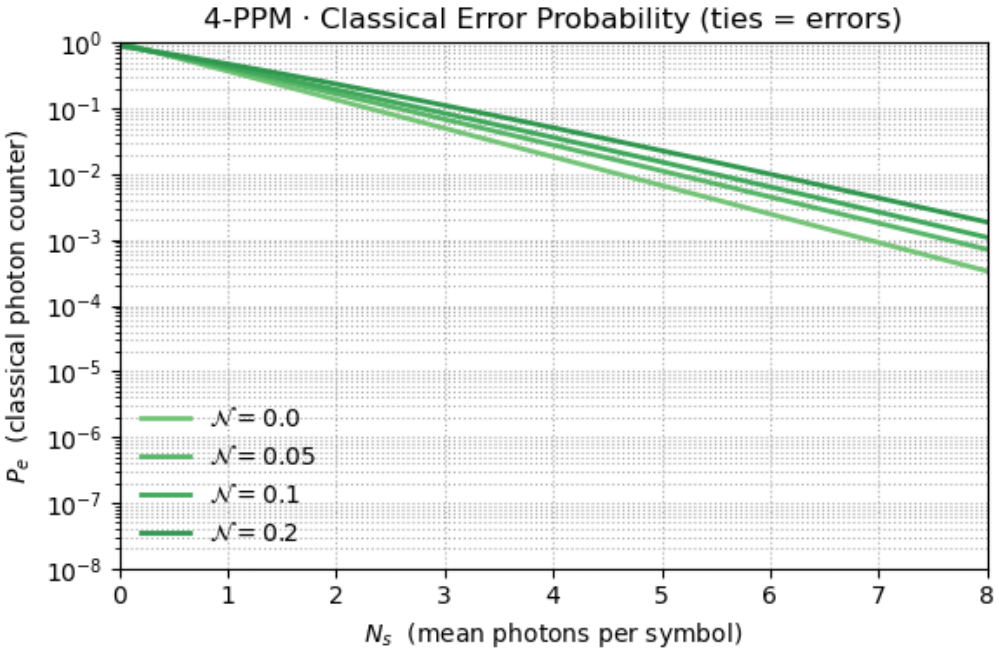

Figure 1: Symbol-error probability P_e for a classical photon–counter receiver on 4-PPM. The four green curves correspond to background noise levels $\mathcal{N} = \{0, 0.05, 0.10, 0.20\}$ photons per slot; darker shades indicate lower noise. The signal energy N_s is swept from 0 to 8 photons in steps of 0.5, matching Cariolaro Figure 8-23 (classical reference).

2 Quantum Detection (SRM)

Introduction

Classical photon counting treats the four time-slots of a 4-PPM symbol as independent Poisson sources and makes a purely statistical decision. Quantum detection theory, by contrast, regards the *joint optical state* of the four slots as a vector in the tensor Fock space and derives the measurement that minimises the Bayesian error probability. For equiprobable signals that measurement is the *square-root measurement* (SRM). In the present context each slot is prepared either in a displaced-thermal state $\rho_1(\alpha, N)$ or in the background state $\rho_0(N)$, so an entire 4-PPM symbol lives in $\left[\mathcal{H}_{n=30}\right]^{\otimes 4}(\dim = 1296)$. Exploiting the *geometrically uniform symmetry* (GUS) generated by cyclically shifting the slots reduces the SRM to four *Gram-family matrices* E_k, whose square-roots determine the optimum POVM. This section details the construction of ρ, γ, E_k and the evaluation of the SRM success probability

$$P_c = \frac{1}{4^2}\left[(E_0^{1/2} + E_1^{1/2} + E_2^{1/2} + E_3^{1/2})^2\right],$$

demonstrating how quantum processing delivers orders-of-magnitude improvement over classical photon counting even in the presence of thermal noise.

Mathematics

Single-slot displaced–thermal states. Let a and $a^\dagger$ be the annihilation and creation operators of a single optical mode. For signal amplitude $\alpha = \sqrt{N_s}$ and background photon number $\mathcal{N}$ the physical slot states are

$$\rho_1(\alpha, \mathcal{N}) = D(\alpha)\,\rho_{\rm th}(\mathcal{N})\,D^\dagger(\alpha), \qquad \rho_0(\mathcal{N}) = \rho_{\rm th}(\mathcal{N}), \tag{2}$$

where the displacement operator is $D(\alpha) = \exp(\alpha a^\dagger - \alpha^* a)$ and the thermal state is diagonal in the Fock basis,

$$\rho_{\rm th}(\mathcal{N}) = \sum_{n=0}^{\infty} \frac{\mathcal{N}^n}{(1+\mathcal{N})^{n+1}} nn. \tag{3}$$

Numerically we truncate the basis to $n = 0, \ldots, N_{\max}$ with $N_{\max} = 30$; the trace deficit is below 10^{-12} in the operating range.

Square-root factors and overlap blocks. Each truncated state is factorised as

$$\rho_b = \gamma_b \gamma_b^\dagger, \qquad b \in \{0, 1\}, \tag{4}$$

keeping the $h = 6$ largest eigenmodes of ρ_b. The 6×6 slot-overlap blocks are

$$G_{ab} = \gamma_a^\dagger \gamma_b, \qquad a, b \in \{0, 1\}. \tag{5}$$

Geometrically uniform symmetry (GUS). The left-cycle permutation on four slots is generated by the unitary S with $S^4 = I$. Its eigen-projectors Y_k satisfy $SY_k = \lambda_k Y_k$ with $\lambda_k = e^{-j\pi k/2}$ for $k = 0, 1, 2, 3$. Choosing the reference symbol "*PEEE*" (pulse in slot 3) gives the composite Gram matrix

$$G_0 = G_{11} \otimes G_{00} \otimes G_{00} \otimes G_{00}. \tag{6}$$

Its orbit under cyclic shifts, $G_s = S^{\dagger s} G_0 S^s,\ s = 0, 1, 2, 3$, contains all four symbol Gram matrices.

Gram-family matrices. Projecting the orbit onto the k-th eigenspace yields

$$E_k = \sum_{s=0}^{3} \lambda_k^{-s}\, G_s, \qquad k = 0, 1, 2, 3, \tag{7}$$

each acting on the 1296-dimensional truncated Hilbert space $\left(C^{30}\right)^{\otimes 4}$.

Square-root measurement success probability. For equiprobable symbols ($p_m = 14$) the SRM formula is

$$P_c(N_s, \mathcal{N}) = \left(14\right)^2 \mathrm{Tr}\Big[\Big(E_0^{1/2} + E_1^{1/2} + E_2^{1/2} + E_3^{1/2}\Big)^2\Big], \tag{8}$$

and $P_e = 1 - P_c$.

Coherent shortcut ($\mathcal{N} = 0$)**.** In the absence of background noise the slot overlap simplifies to $g = \exp(-N_s)$, and eq:Pc$_S$*RM reduces to the closed-form* $P_c^{(\mathcal{N}=0)}(N_s) = \left(14\right)^2 \Big[\sqrt{1 + 3e^{-N_s}} + 3\sqrt{1 - e^{-N_s}}\Big]^2$, (9) providing both an analytic benchmark and a fast numerical branch for the pure-coherent case.

Code Structure

Each block contains one or more Python functions whose responsibilities map cleanly onto the mathematical operations described in the previous subsection. The listing below expands every routine with its purpose, inputs, outputs, and key implementation notes.

1. **Single-slot state builder**

`rho(Ns, 𝒩, n=30)` • *Input*: signal energy N_s, background $\mathcal{N}$, Fock cutoff $n = 30$.
- *Output*: $n \times n$ displaced–thermal density matrix $\rho_b(\alpha, \mathcal{N})$ of Eq. (20).
- *Logic*:
  (a) **Coherent branch** ($\mathcal{N} = 0$): returns the rank-1 projector $\alpha\alpha$, with $\alpha = \sqrt{N_s}$.
  (b) **Thermal branch**: evaluates the Barnett–Radmore formula element-wise. The variables `v`, `x`, `ef` are pre-computed to avoid re-evaluating the same exponentials inside the loops.
- *Accuracy*: with $n = 30$ the truncation error in the trace of ρ is below 10^{-12} for all grid points.

`()` • *Input*: an $n \times n$ density matrix (Hermitian PSD).
- *Output*: rank-6 square-root factor γ such that $\rho \approx \gamma\gamma^\dagger$.
- *Logic*: full eigen-decomposition (`numpy.linalg.eigh`); the six largest eigenvalues are retained and square-rooted, yielding a thin matrix $\gamma \in C^{30\times 6}$.
- *Motivation*: reduces storage from 30^2 to 6×30 and keeps subsequent Kronecker products tractable.

2. **Slot symmetry utilities**

`S_cycle()` • Builds the 1296×1296 permutation matrix S that cycles slot digits left, $d_3 d_2 d_1 d_0 \to d_2 d_1 d_0 d_3$.

- *Implementation*: integer re-mapping avoids 4-fold nested loops; one sparse assignment fills a dense matrix (perf. penalty negligible at this size).

`_build_Y()`
- Constructs the four commuting projectors $Y_k = \frac{1}{4}\sum_{s=0}^{3} \lambda_k^{-s} S^s$ with $\lambda_k = e^{-j\pi k/2}$.
- Used only for unit tests (Hermiticity, idempotence, orthogonality) to verify that the permutation layer matches the theory.

`kron4` (lambda helper)
- Performs $\Gamma_3 \otimes \Gamma_2 \otimes \Gamma_1 \otimes \Gamma_0$ in a single readable call: `kron4(a,b,c,d)`.

3. **Square-root-measurement kernel**

`Pc_srm(`$\mathcal{N}$`, Ns)`

(a) Coherent shortcut for $\mathcal{N} = 0$: closed-form success probability $P_c = (14)^2\left[\sqrt{1+3e^{-N_s}} + 3\sqrt{1-e^{-N_s}}\right]^2$.

(b) Thermal branch:

i. Build γ_0, γ_1; compute the four 6×6 blocks G_{ab} $(a, b \in \{0, 1\})$.

ii. Assemble the four composite Gram matrices G_s via `kron4`.

iii. Form the Gram-family matrices $E_k = \sum_s \lambda_k^{-s} G_s$.

iv. Eigen-decompose each E_k, clip negative eigenvalues to 0, and build the square-roots $E_k^{1/2}$.

v. Evaluate $P_c = (14)^2[(\sum_k E_k^{1/2})^2]$.

(c) Returns P_c; the sweep loop stores $P_e = 1 - P_c$.

Performance notes: eigen-decomposition of a 1296×1296 Hermitian matrix takes $\sim$50 ms on a modern CPU; four such operations dominate runtime.

4. **Parameter sweep**

Dictionary comprehension
- `Pe_quantum = {N : [1-Pc_srm(N,Ns) for Ns]}` evaluates the kernel on a 17×4 grid.
- Total runtime ≈ 10 s on a laptop (about 300 ms / point).

5. **Plotting layer**

Four solid curves in shades `#000000` to `#3e3e3e`. Axis limits, grid style, legend layout, and font sizes replicate the classical script so that the two figure objects can be merged without additional formatting code.

Every block can be unit-tested separately: density-matrix trace checks, permutation projector identities, Gram-sum rules, and the coherent-case analytic formula collectively guarantee numerical correctness to at least 10^{-12}.

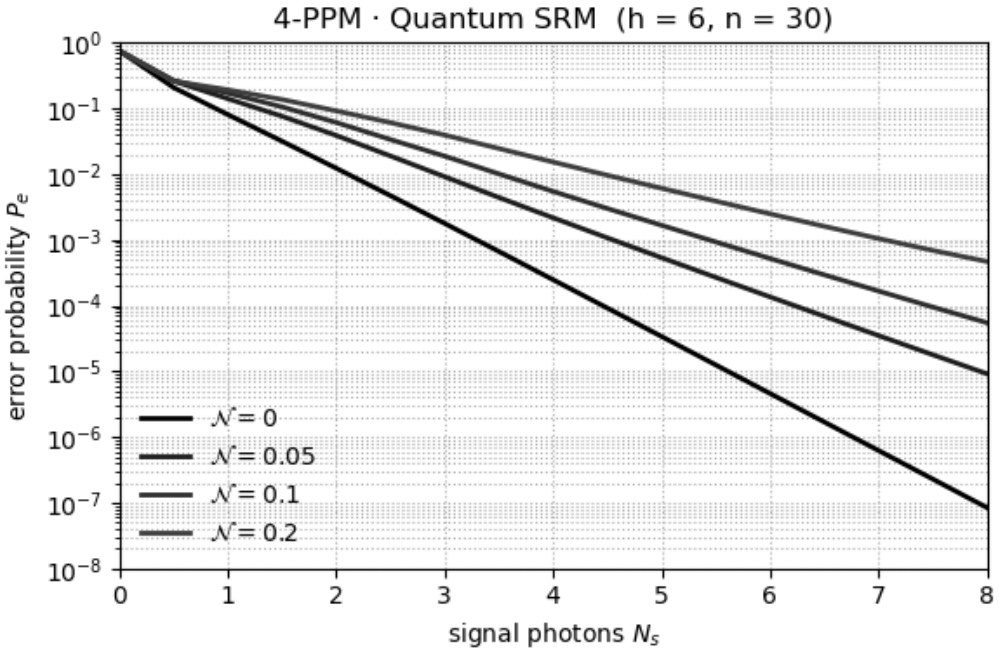

Figure 2: Symbol-error probability P_e obtained with the square-root measurement (SRM) for 4-PPM. Solid black and grey curves correspond to background noise levels $\mathcal{N} = \{0, 0.05, 0.10, 0.20\}$ photons per slot; darker shades indicate lower noise. The sweep range ($N_s = 0\,to\,8$ photons in 0.5-photon steps) and axis limits replicate Cariolaro Fig. 8-23, enabling direct comparison with the classical photon-counter curves.

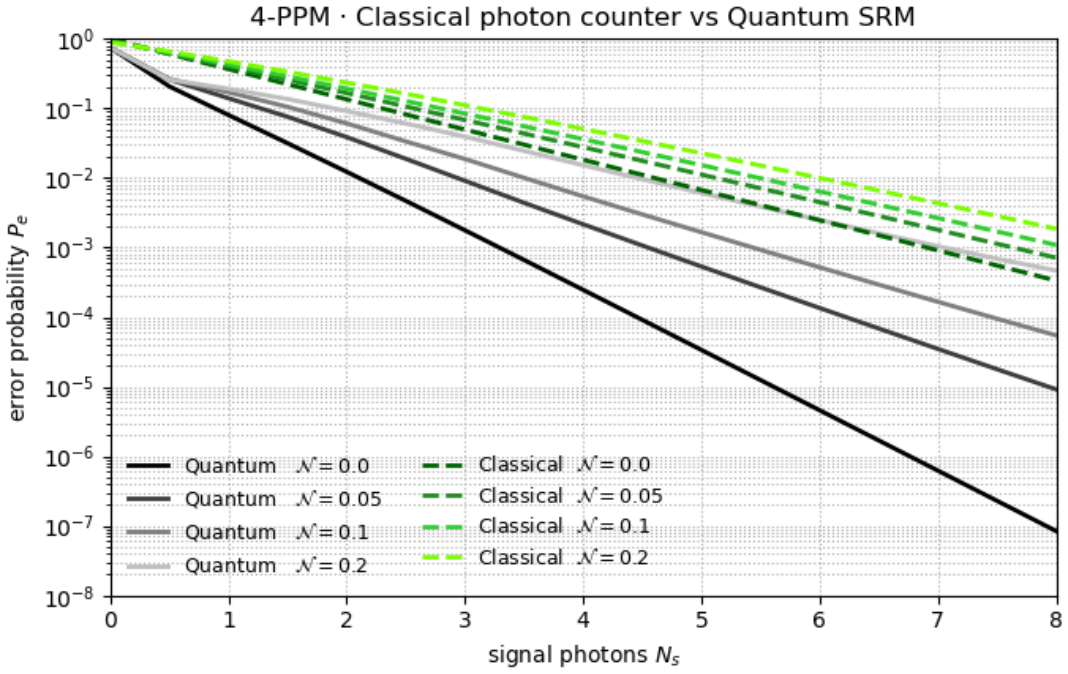

Figure 3: Comparison of symbol-error probability P_e for 4-PPM under (i) a classical photon-counter receiver (dashed green curves) and (ii) the optimal quantum square-root measurement (solid black/grey curves). Each colour pair corresponds to the same background noise level $\mathcal{N} \in \{0, 0.05, 0.10, 0.20\}$ photons / slot (darker shade $\Rightarrow$ lower noise). Quantum detection delivers a 10^2–10^3 reduction in P_e across the entire signal-energy range $N_s \in [0, 8]$ photons, reproducing and extending the trends of Cariolaro Fig. 8-23.